Renesas USB MCU

R01AN2236JJ0111
Rev.1.11
Sep 30, 2015

USB Host Human Interface Devices Class Driver (HHID) Using Firmware Integration Technology Modules

要旨

本資料は、USB Host Human Interface Devices Class Driver Firmware Integration Technology を使用したサンプルファームウェアの説明資料です。以降、本サンプルファームウェアを HHID と記述します。

実際のソフトウェア開発時には、必ず”USB Host Human Interface Device Class Driver (HHID) Firmware Integration Technology アプリケーションノート”(Document No.R01AN2028JJ)、”USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート”(Document No. R01AN2025JJ)及び各マイコンのユーザーズマニュアル(ハードウェア編)と併用してご利用ください。

対象デバイス

RX63N/RX631 グループ

RX64M グループ

RX71M グループ

本プログラムは Renesas Starter Kit (RSK)を使って動作確認を行っています。

目次

1. はじめに 2
2. ソフトウェア構成 6
3. サンプルアプリケーション 7
4. クラスドライバ概要 15
5. e² studio 用プロジェクトを CS+で使用する場合 16

1. はじめに

1.1 機能概要

HHID は、USB ヒューマンインターフェースデバイスクラス仕様（以降 HID と記述）に準拠し、HID デバイスと通信を行うことが可能です。

HHID の機能を以下に示します。

- マウスまたはキーボード接続時、HID クラスのデータ転送を行う。

1.2 FIT モジュール構成

HHID は以下の FIT モジュールとサンプルアプリケーションで構成されています。

Table 1-1 FIT モジュール構成

FIT モジュール名	フォルダ名	バージョン
RX Family Board Support Package Module Using Firmware Integration Technology	r_bsp	2.80
Renesas USB MCU USB Basic Host and Peripheral firmware Firmware Integration Techology	r_usb_basic	1.11
Renesas USB MCU USB Host Human Interface Devices Class Driver(HID) Firmware Integration Techology	r_usb_hhid	1.11

各 FIT モジュールの詳細は、関連ドキュメントを参照してください。また、本サンプルファームウェアで使用している FIT モジュールの最新バージョンは下記のホームページよりダウンロードが可能です。

ルネサスエレクトロニクスホームページ http://japan.renesas.com/

1.3 動作確認済環境

HHID の動作確認環境を以下に示します。

＜評価ボード＞

Renesas Starter Kit+ for RX63N（RSK+RX63N）：型名:R0K50563NC000
RX63N グループ ルネサスマイコン開発スタータキット ルネサスエレクトロニクス製

Renesas Starter Kit+ for RX64M（RSK+RX64M）：型名:R0K50564MC001BR
RX64M グループ ルネサスマイコン開発スタータキット ルネサスエレクトロニクス製

Renesas Starter Kit+ for RX71M（RSK+RX71M）：型名: R0K5RX71MC010BR
RX71M グループ ルネサスマイコン開発スタータキット ルネサスエレクトロニクス製

＜開発環境＞

a) 統合環境 e² studio ルネサスエレクトロニクス製
b) RX ファミリ用 C/C++コンパイラパッケージ Ver2.03.00 ルネサスエレクトロニクス製
c) E1 エミュレータまたは E20 エミュレータ ルネサスエレクトロニクス製

＜その他＞

a) エミュレータ用ホスト PC（Windows® 7、Windows® 8、Windows® 8.1）
b) HID デバイス(キーボードまたはマウス)
c) USB ケーブル
d) ユーザケーブル（E1 エミュレータまたは E20 エミュレータに同梱）
e) エミュレータ用ケーブル（E1 エミュレータまたは E20 エミュレータに同梱）

1.4 セットアップ

1.4.1 ハードウェア

HHID の動作環境例をFigure 1-1に示します。評価ボードのセットアップ、エミュレータなどの使用方法については各取扱説明書を参照してください。

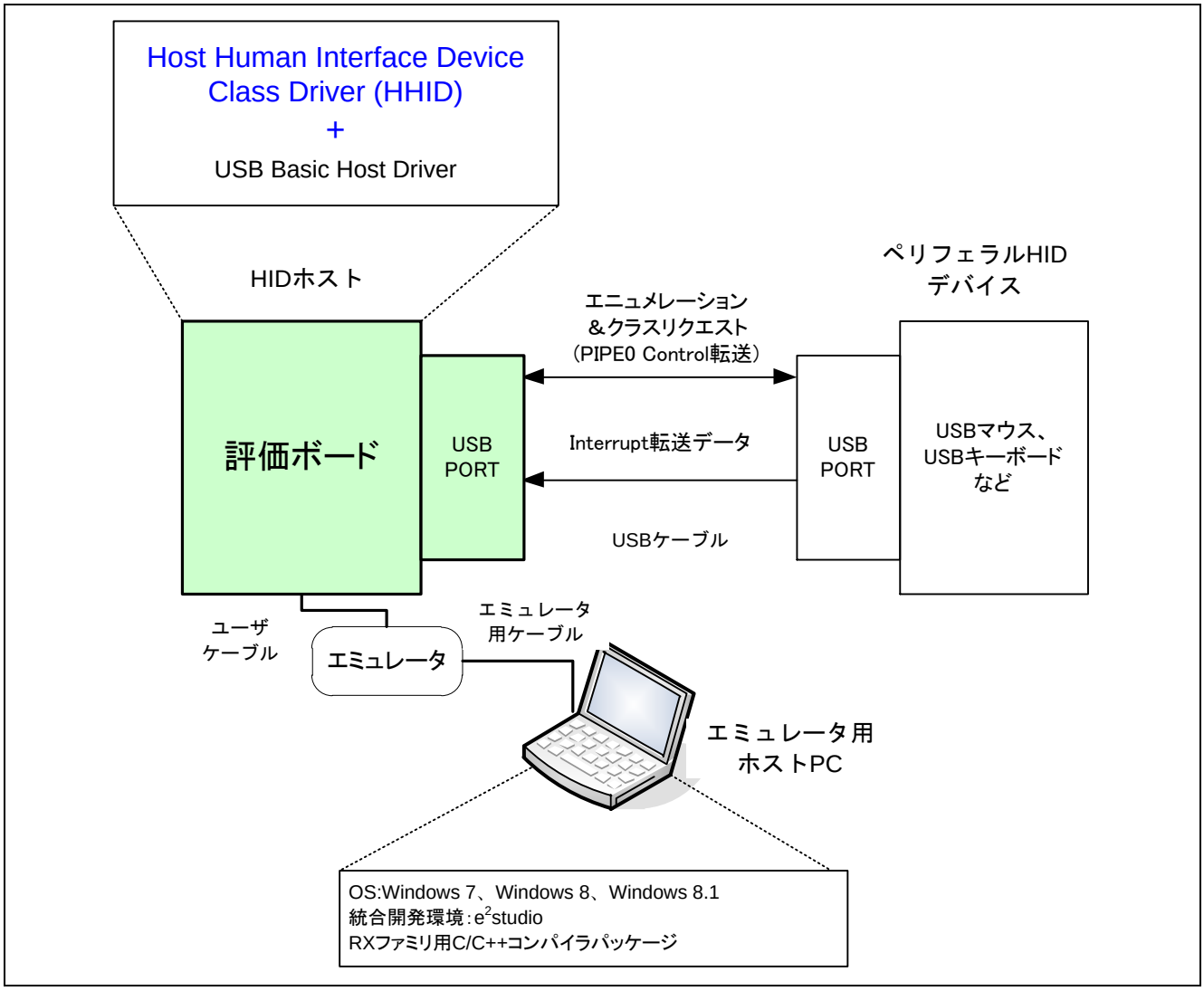


Figure 1-1 動作環境例

動作確認済みの評価ボードをTable 1-2に示します。

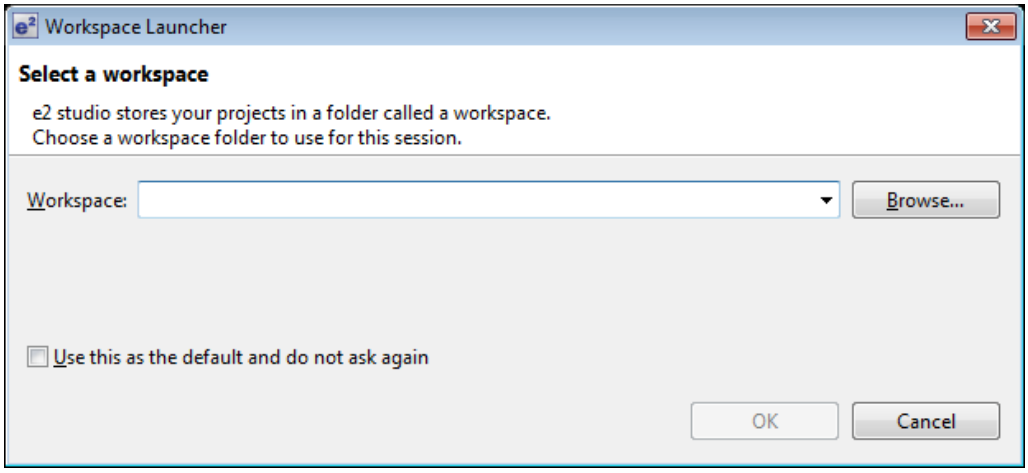
Table 1-2 HHID 動作確認済みの評価ボード

マイコン	評価ボード
RX63N	RSK+RX63N
RX64M	RSK+RX64M
RX71M	RSK+RX71M

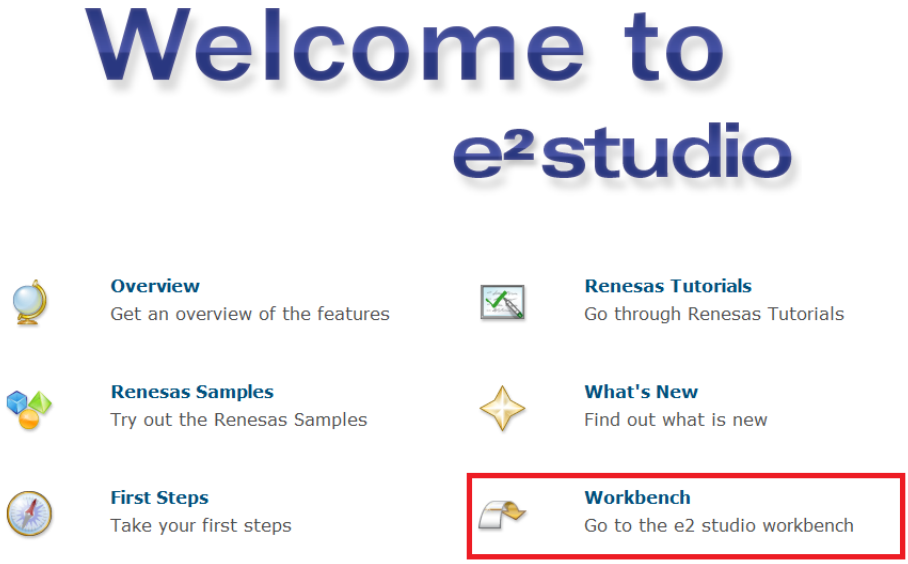
1.4.2 ソフトウェア

(1). e² studio を起動

a) e² studio を起動してください。
b) はじめて e² studio を起動する場合、Workspace Launcher ダイアログが表示されますので、プロジェクトを格納するためのフォルダを指定してください。

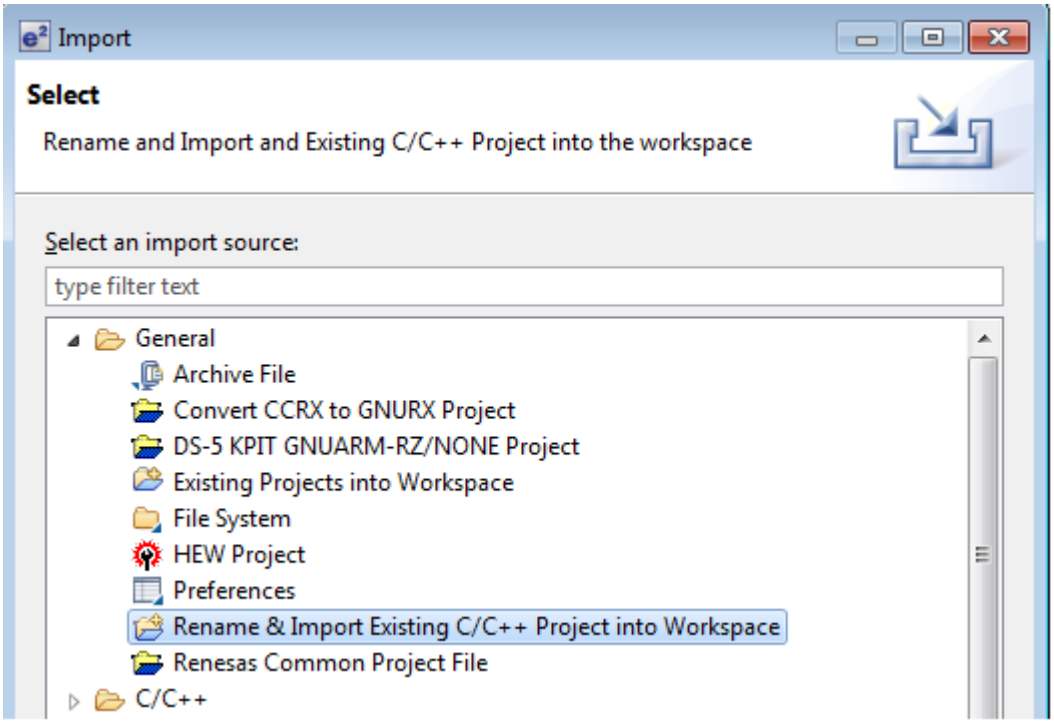

Workspace Launcher
Select a workspace
e2 studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.
Workspace:
Browse...
Use this as the default and do not ask again
OK
Cancel

c) Workbench アイコンをクリックしてください。

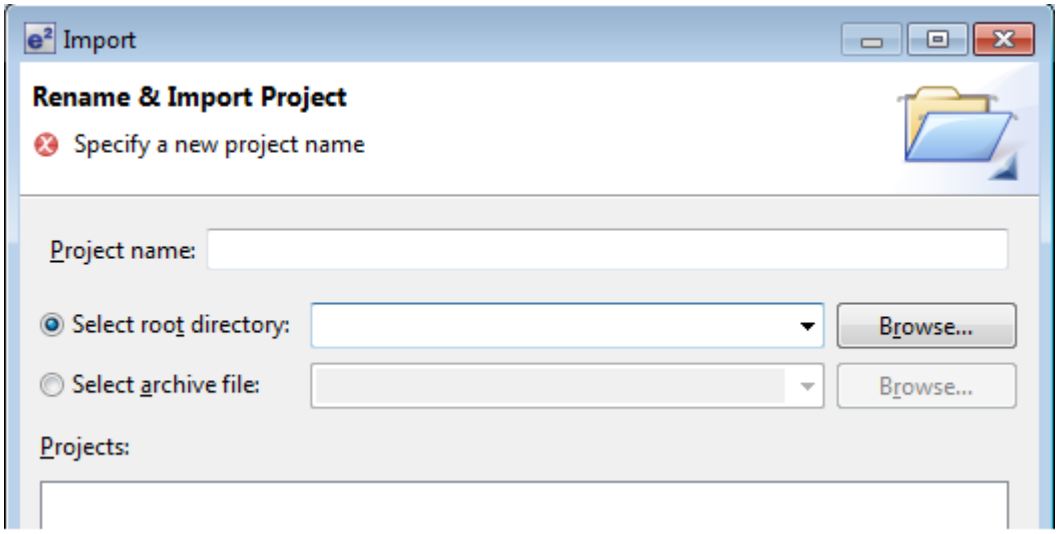


(2). ワークスペースへのプロジェクトの登録

a) [File] --> [Import]を選択してください。
b) General => Rename & Import Existing C/C++ Project into Workspace を選択してください。



プロジェクトファイル".cproject"が格納されたフォルダを”Select root directory”に入力してください。



c) “Finish”をクリック

プロジェクトのワークスペースへのインポートが完了しました。同様の方法で他のプロジェクトを同一のワークスペースへインポートすることができます。

(3). “Build”ボタンをクリックし、実行プログラムを生成してください。

(4). デバッガへの接続を行い、実行プログラムをダウンロードしてください。“Run”ボタンをクリックすると、プログラムが実行されます。

2. ソフトウェア構成

2.1 モジュール構成

HHID は、HID クラスドライバとマウス、キーボードのデバイスドライバから構成されます。接続された HID デバイスからデータを受け取ると、HCD を介して User application(APL)に結果を通知します。また、APL からデータ転送要求があった場合、HCD を介して HID デバイスに通知します。

Figure 2-1に HHID のモジュール構成、Table 2-1にモジュール機能概要を示します。

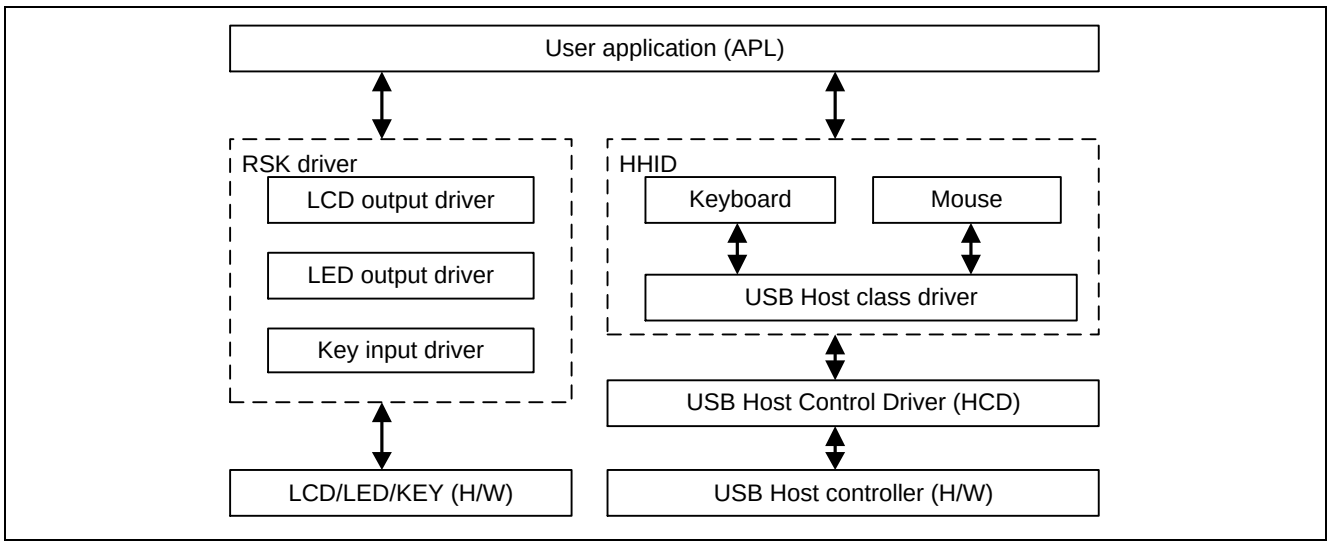


Figure 2-1 モジュール構成図

Table 2-1 モジュール機能概要

モジュール名	機能概要
APL	サンプルアプリケーションプログラム ・ HID デバイスとの通信開始、スイッチ操作によるサスペンド/レジューム制御を行う ・ HID デバイスから受信したレポート情報を LCD に表示する
RSK driver	RSK 上の各周辺機能を使用するためのサンプルアプリケーション
HHID (r_usb_hhid)	HID クラスドライバ ・ HID デバイスからの要求を解析する ・ HCD を介して APL のスイッチ操作情報を HID デバイスに通知する
HCD (r_usb_basic)	USB Host H/W 制御ドライバ

3. サンプルアプリケーション

3.1 アプリケーション仕様

HHID のサンプルアプリケーション(以降、APL)は、RSK 上に実装されたスイッチ、LED および液晶表示器(LCD）を使用します。
APL の主な機能を以下に示します。

1) HID デバイス(マウス/キーボード)とのデータ転送

接続された HID デバイスとのデータ転送を行います。USB Hub を使用すれば最大 3 つの HID デバイスを接続することができ、最大 3 つの HID デバイスとのデータ転送を同時におこないます。

2) HID デバイスから受信したデータの表示

a) マウス接続時、X 軸、Y 軸の移動量を LCD に数値表示を行い、クリックボタン押下で LED 点灯を行います。
b) キーボード接続時、入力されたキーデータを LCD に 1 文字表示します。

3) HID デバイスのサスペンド/レジューム実行

a) RSK 上のスイッチ 3（SW3:状態変更スイッチ）押下により、接続された HID デバイスに対してサスペンド、レジュームを行います。（注 1）
b) リモートウェイクアップ対応デバイスからのリモートウェイクアップ信号により、レジュームを行います。（注 1）

（注1）サスペンド/レジュームは RSK の USB ポートに直接 HID デバイスを接続した場合のみ行います。
HUB を介して接続された HID デバイスに対するサスペンド/レジュームはサポートしていません。

3.1.1 スイッチ

APL は HID デバイスと接続後、データ転送を開始します。データ転送中にスイッチ入力を行うことで以下の動作を行います。

1. データ転送状態時に SW3 を押下することで HID デバイスをサスペンド状態に移行する。
2. HID デバイスがサスペンド状態の時、SW3 を押下することでサスペンド状態を解除する。

Table 3-1にスイッチ入力の仕様を示します。

Table 3-1 スイッチ入力の仕様

スイッチ名称	スイッチ番号	動作内容
状態変更スイッチ	スイッチ 3	HID デバイスの状態を変更する ・ データ転送状態：サスペンド状態に移行 ・ サスペンド状態：データ転送状態に移行

※RSK のスイッチと MCU ピン接続は、使用している RSK の取扱説明書、各 MCU のユーザーズマニュアルを参照してください。

3.1.2 MCU 端子設定

このアプリケーションプログラムでは、VBUS 端子および OVRCUR 端子の設定は r_usb_hhid_apl.c ファイル内の usb_mcu_init 関数で行われています。

3.1.3 表示情報

APL は、HID デバイスの接続状態、及び接続された HID デバイスから受信データ内容を LCD に表示します。

マウス接続時 : X/Y 軸の移動量（-127～127）を LCD に表示、クリック時 LED 点灯
キーボード接続時 : 最後に入力されたキーデータを LCD に表示

HID デバイスから受信データの内容が NULL（キーボード非押下、マウスの X/Y 軸移動なし）の場合、LCD 表示を更新しません。

3.2 アプリケーション処理概要

APL は、初期設定、メインループの 2 つの部分から構成されます。以下にそれぞれの処理概要を示します。

3.2.1 初期設定

初期設定では、MCU の端子設定、USB ドライバの設定、USB コントローラの初期設定を行います。

3.2.2 メインループ

このメインループでは、以下の処理を行います。

1. APL は、ステートとそのステートに関連するイベントにより管理を行っています。APL では、まず、接続されたデバイスごとにステート (Table 3-2参照)の確認を行います。なお、このステートは、APL が管理する構造体(“3.2.3 ステートとイベントの管理”参照)のメンバに格納されています。
2. APL は、そのステートに関連するイベント(Table 3-3参照)の確認を行い、そのイベントに応じた処理を行います。そのイベント処理後、APL は、必要に応じてステートを変化させます。なお、このイベントは、APL が管理する構造体(“3.2.3 ステートとイベントの管理”参照)のメンバに格納されています。

以下に、APL の処理概要を示します。

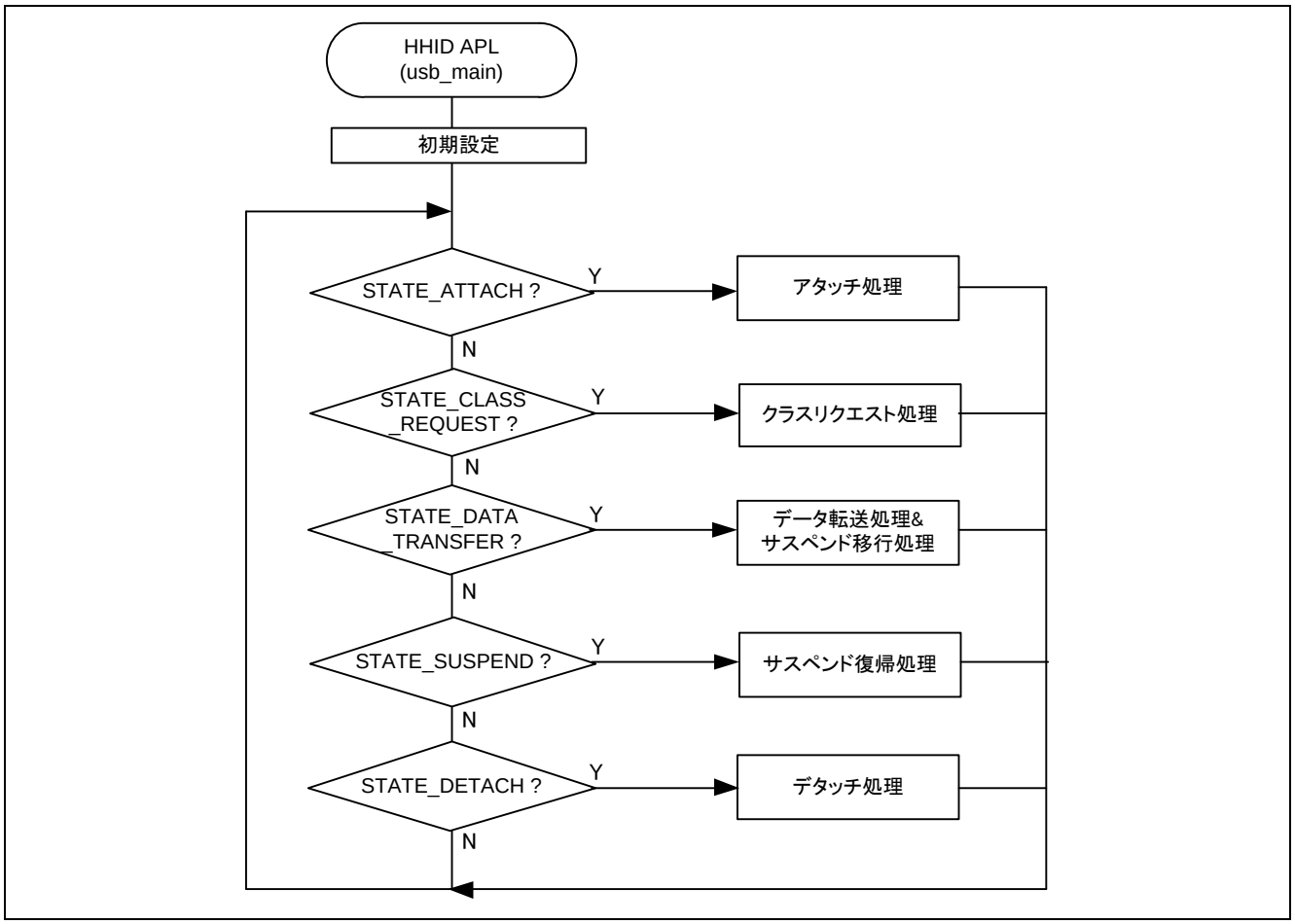


Figure 3-1 メインループ処理

Table 3-2 ステート一覧

ステート	ステート処理概要	関連イベント
STATE_ATTACH	アタッチ処理	EVENT_CONFIGURD
STATE_CLASS_REQUEST	クラスリクエスト処理	EVENT_CLASS_REQUEST_START
		EVENT_USB_TRANSFER_COMPLETE
STATE_DATA_TRANSFER	データ転送& サスペンド移行処理	EVENT_USB_TRANSFER_START
		EVENT_USB_TRANSFER_COMPLETE
		EVENT_SWITCH_INPUT
STATE_SUSPENDED	サスペンド復帰処理	EVENT_SWITCH_INPUT
STATE_DETACH	デタッチ処理	'--

Table 3-3 イベント一覧

イベント	概要
EVENT_CONFIGURD	USB デバイス接続完了
EVENT_CLASS_REQUEST_START	クラスリクエスト要求
EVENT_CLASS_REQUEST_COMPLETE	クラスリクエスト完了
EVENT_USB_TRANSFER_START	データ転送要求
EVENT_USB_TRANSFER_COMPLETE	データ転送完了
EVENT_SWITCH_INPUT	スイッチ入力
EVENT_NONE	イベント無し

3.2.3 ステートとイベントの管理

ステートとイベントは、以下の構造体のメンバ(*state, event[]*)によって管理されています。この構造体は、APL が用意している構造体で、接続する HID デバイスのデバイスアドレスをもとに管理されます。

```
typedef struct DEV_INFO                 /* Structure for HID device control */
{
    uint16_t    state;                  /* State for application */
    uint16_t    event_cnt;              /* Event count */
    uint16_t    event[EVENT_MAX];       /* Event. */
    uint16_t    pipe;                   /* Use pipe no. */
    uint16_t    report_length;          /* USB receive data size.*/
    uint16_t    class_request;          /* Class Request Type. */
    uint16_t    protocol;               /* HID Protocol(Mouse/Keybord/none) */
}
DEV_INFO_t;
```

[Note]

イベント取得処理で、取り出すイベントが存在しない場合、“EVENT_NONE”が返されます。

以下にステートごとの処理概要を示します。

1. アタッチ処理 (STATE_ATTACH)

== 概要 ==

このステートでは、RSK に HID デバイスが ATTACH され、Enumeration が完了したことを APL に通知する処理を行い、ステートを STATE_CLASS_REQUEST に変更します。

== 内容 ==

① APL では、はじめに初期化関数がステートを STATE_ATTACH にセットし、イベントに EVENT_NONE をセットします。
② HID デバイスが接続されるまで STATE_ATTACH 状態が継続され、hid_connect_wait()がコールされます。HID デバイスが接続され、Enumeration が完了するとコールバック関数 hid_configured が USB ドライバよりコールされ、イベント EVENT_CONFIGURD を発行します。なお、このコールバック関数は、USB_HCDREG_t 構造体のメンバ devconfig に設定した関数です。
③ イベント EVENT_CONFIGURD では、hid_connect_wait 関数内で以下の処理が行われ、次のループ処理でクラスリクエスト処理を開始します。
   a) 接続された HID デバイスの情報を特定の変数(hid_dev_info)に設定
   b) ステートを STATE_CLASS_REQUEST に変更し、イベント EVENT_CLASS_REQUEST_START を発行します。

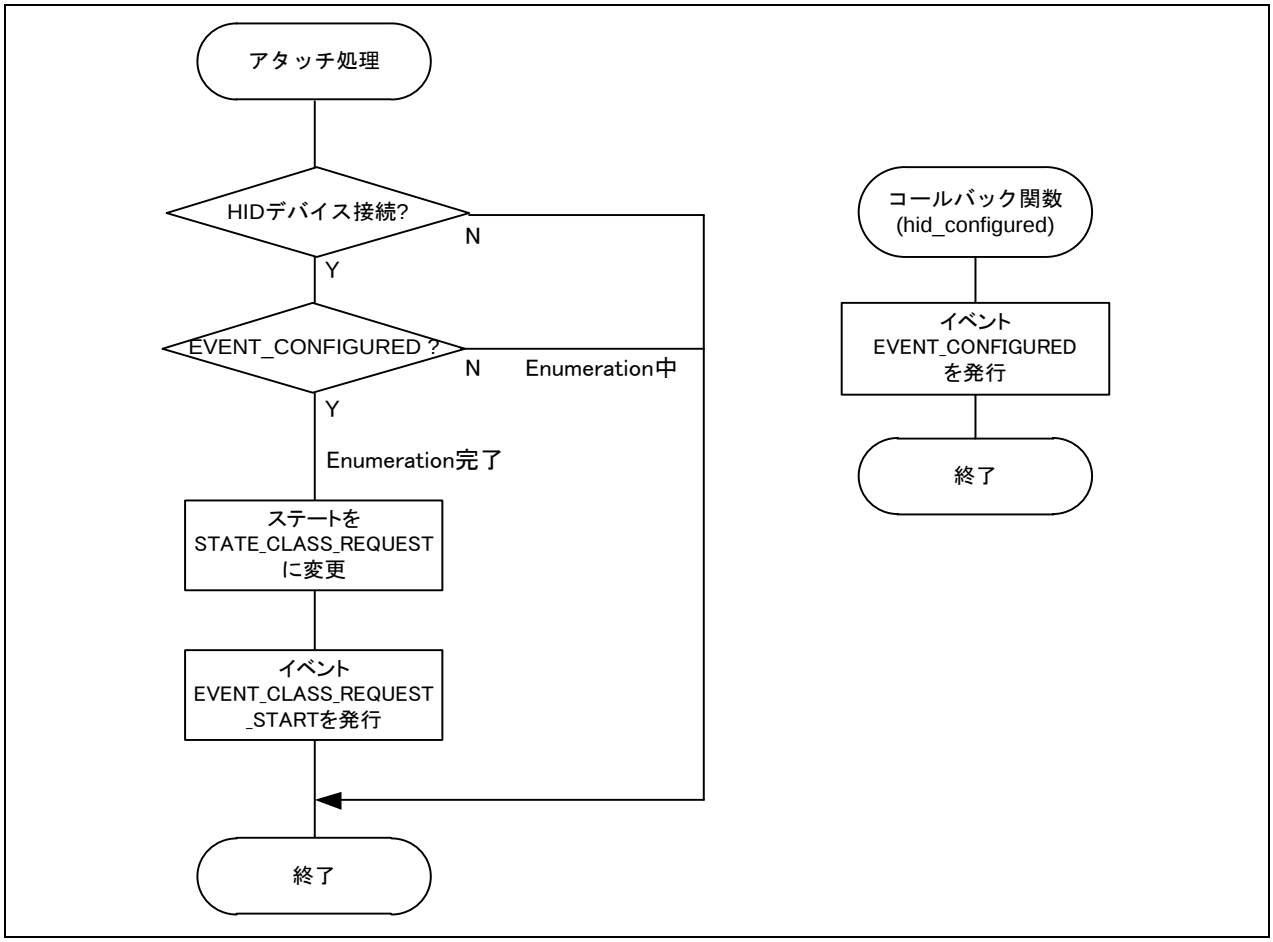


Figure 3-2 アタッチ処理概略フロー

2. クラスリクエスト処理 (STATE_CLASS_REQUEST)

== 概要 ==

このステートでは、HID デバイスに対しクラスリクエストを送信する処理を行います。クラスリクエスト送信が完了するとステートを STATE_DATA_TRANSFER に変更します。

== 内容 ==

① HID デバイスとの Enumeration が完了するとステートが STATE_CLASS_REQUEST に遷移します。このステートでは、はじめに EVENT_CLASS_REQUEST_START が処理され、USB ドライバに対しクラスリクエスト送信要求を行います。

② クラスリクエスト送信処理が完了するとコールバック関数 hid_class_request_complete がコールされます。このコールバック関数では、イベント EVENT_CLASS_REQUEST_COMPLETE を発行します。

③ EVENT_CLASS_REQUEST_COMPLETE では、上記①で送信したクラスリクエストによって以下の通りに処理が分かれます。

a) マウスに対しクラスリクエスト set_protocol を送信した場合、ステートを STATE_DATA_TRANSFER に変更し、イベント EVENT_USB_TRANSFER_START を発行します。次のループ処理でデータ転送処理を開始します。

b) キーボードに対しクラスリクエスト set_protocol を送信した場合、イベント EVENT_CLASS_REQUEST_START を発行し、クラスリクエスト set_protocol 送信処理を開始します。

c) クラスリクエスト set_report を送信した場合、ステートを STATE_DATA_TRANSFER に変更し、イベント EVENT_USB_TRANSFER_START を発行します。

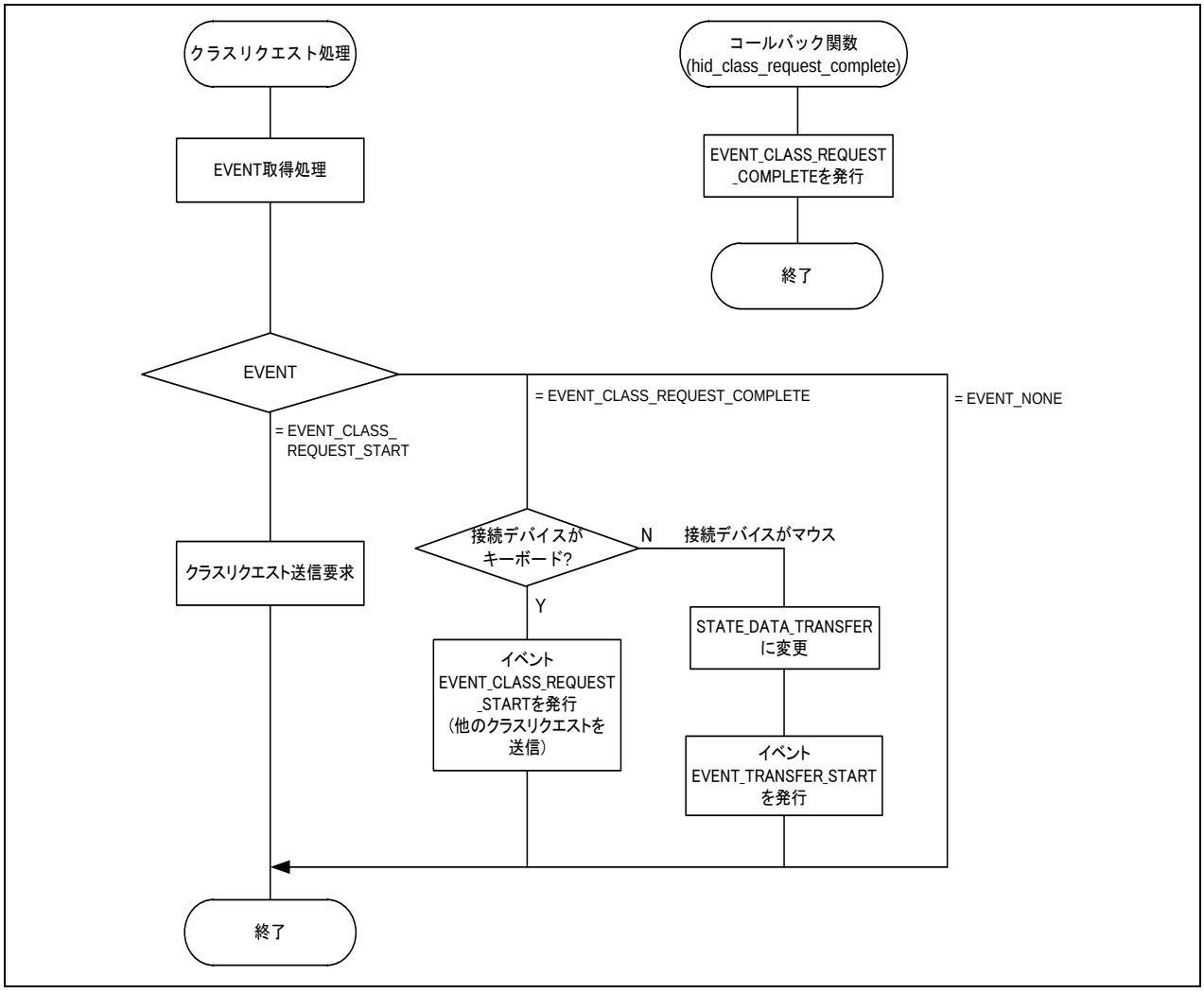


Figure 3-3 クラスリクエスト処理概略フロー

3. データ転送処理&サスペンド移行処理 (STATE_DATA_TRANSFER)

== 概要 ==

このステートでは、HID デバイスからのデータ受信処理およびそのデータの LCD 表示処理等を行います。その他、スイッチ押下による HID デバイスに対するサスペンド信号送信処理も行います。

== 内容 ==

① HID デバイスに対するクラスリクエストの送信が完了するとステートは STATE_DATA_TRANSFER に遷移します。このステートでは、hid_data_transfer 関数内の EVENT_USB_TRANSFER_START が処理され、USB ドライバに対しデータ転送処理要求を行います。
② データ転送処理が完了するとコールバック関数 hid_receive_complete がコールされます。このコールバック関数は、イベント EVENT_USB_TRANSFER_COMPLETE を発行します。
③ EVENT_USB_TRANSFER_COMPLETE では、hid_receive_data 関数は、受信データの LCD 表示等を行った後、イベント EVENT_USB_TRANSFER_START を発行します。
④ 上記①から③までのデータ転送処理中に RSK 上のスイッチ 3 が押されるとステートを STATE_SUSPEND に変更し、R_usb_hhid_ChangeDeviceState 関数を使って接続された HID デバイスに対し SUSPEND 信号を送信します。

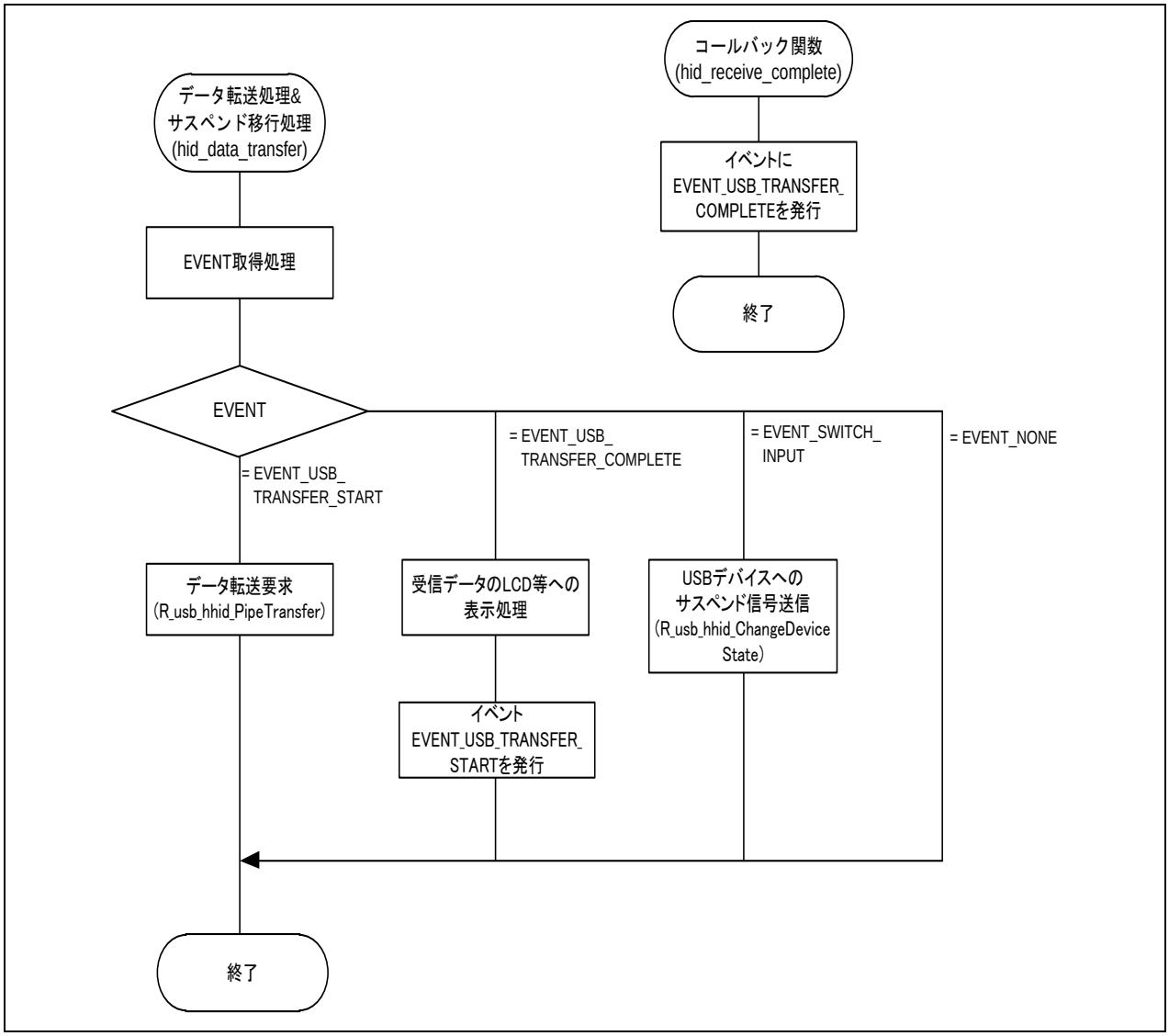


Figure 3-4 データ転送処理 & サスペンド移行処理概略フロー

4. サスペンド復帰処理 (STATE_SUSPENDED)

== 概略 ==

このステートでは、サスペンド状態の HID デバイスに対し RESUME 信号を送信し、サスペンド状態からの復帰処理を行います。RESUME 信号の送信完了は、コールバック関数 hid_resume_complete がコールされることにより APL に通知されます。

== 内容 ==

① APL は、RSK 上のスイッチ 3 の押下を監視します。
② スイッチ 3 が押下されると、hid_suspended 関数は R_usb_hhid_ChangeDeviceState 関数を使って、接続された HID デバイスに対し RESUME 信号を送信します。
③ RESUME 信号の送信が完了すると USB ドライバよりコールバック関数 hid_resume_complete がコールされます。このコールバック関数は、ステートを STATE_DATA_TRANSFER に変更し、イベント EVENT_USB_TRANSFER_START を発行します。

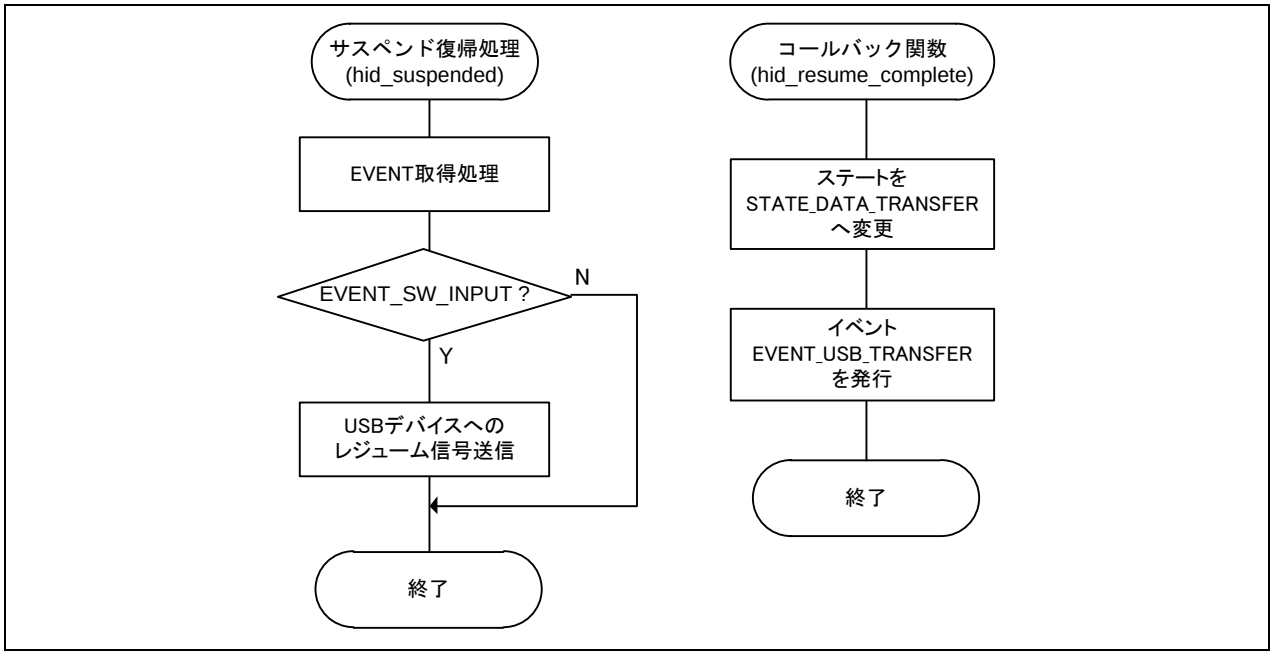


Figure 3-5 サスペンド復帰処理概略フロー

5. デタッチ処理 (STATE_DETACH)

接続された HID デバイスがデタッチされると USB ドライバよりコールバック関数 hid_detach がコールされます。このコールバック関数は、ステートを STATE_DETACH に変更します。STATE_DETACH では、特定の変数(hid_dev_info)にクリア処理、ステートを STATE_ATTACH に変更するなどの処理を行います。

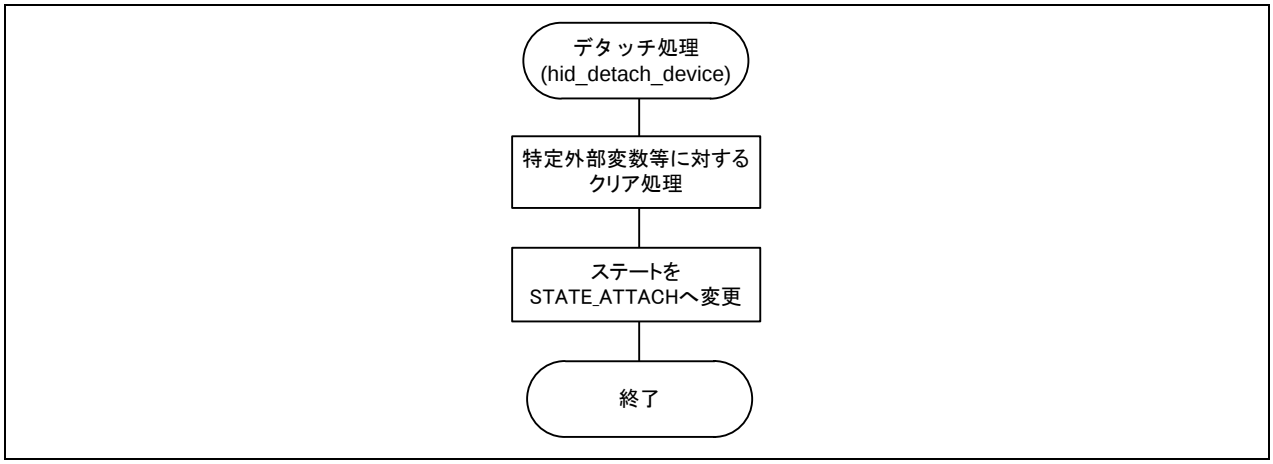


Figure 3-6 デタッチ処理概略フロー

3.3 注意事項

複数の HID デバイスを接続する場合は、コンパイルするファイルを以下にとおりに変更してください。

r_usb_hhid_apl.c	→	r_usb_hhid_apl_mult.c
r_usb_hhid_apl_data.c	→	r_usb_hhid_apl_data_mult.c
r_usb_hhid_defep.c	→	r_usb_hhid_defep_mult.c

4. クラスドライバ概要

4.1 クラスリクエスト（ホストからデバイスへの要求）

HHID がサポートしているクラスリクエストをTable 4-1に示します。

Table 4-1 対応する基本リクエスト及び HID クラスリクエスト

リクエスト	コード	説明
GET_REPORT	0x01	HID デバイスにレポートを要求する
SET_REPORT	0x09	HID デバイスにレポートを通知する
GET_IDLE	0x02	HID デバイスに Duration 時間を要求する
SET_IDLE	0x0A	HID デバイスに Duration 時間を通知する
GET_PROTOCOL	0x03	HID デバイスにプロトコルを要求する
SET_PROTOCOL	0x0B	HID デバイスにプロトコルを通知する
GET_REPORT_DESCRIPTOR	Standard	レポートディスクリプタを要求する
GET_HID_DESCRIPTOR	Standard	HID ディスクリプタを要求する

4.2 データフォーマット

キーボードまたはマウスからインタラプト IN 転送で受信するデータの Boot プロトコル時のフォーマットを以下に示します。

Table 4-2 受信データフォーマット(Boot プロトコル)

offset	キーボード (8 Bytes)	マウス (3 Bytes)
0 (Top Byte)	Modifier keys	b0：Button 1 b1：Button 2 b2：Button 3 b3-b7：Reserved
+1	Reserved	X displacement
+2	Keycode 1	Y displacement
+3	Keycode 2	―
+4	Keycode 3	―
+5	Keycode 4	―
+6	Keycode 5	―
+7	Keycode 6	―

ご注意

Report プロトコル時のデータ転送で用いるデータのフォーマットはレポートディスクリプタに従う必要があります。

本ドライバではレポートディスクリプタの取得と解析は行わず、インタフェースプロトコルコードが“キーボード”か“マウス”のいずれかに従って、レポートフォーマットを決定しています。システムの仕様にあわせてユーザーが変更してください。

5. e² studio 用プロジェクトを CS+で使用する場合

HHID のプロジェクトは、統合開発環境 e²studio で作成されています。HHID を CS+で動作させる場合は、下記の手順にて読み込んでください。

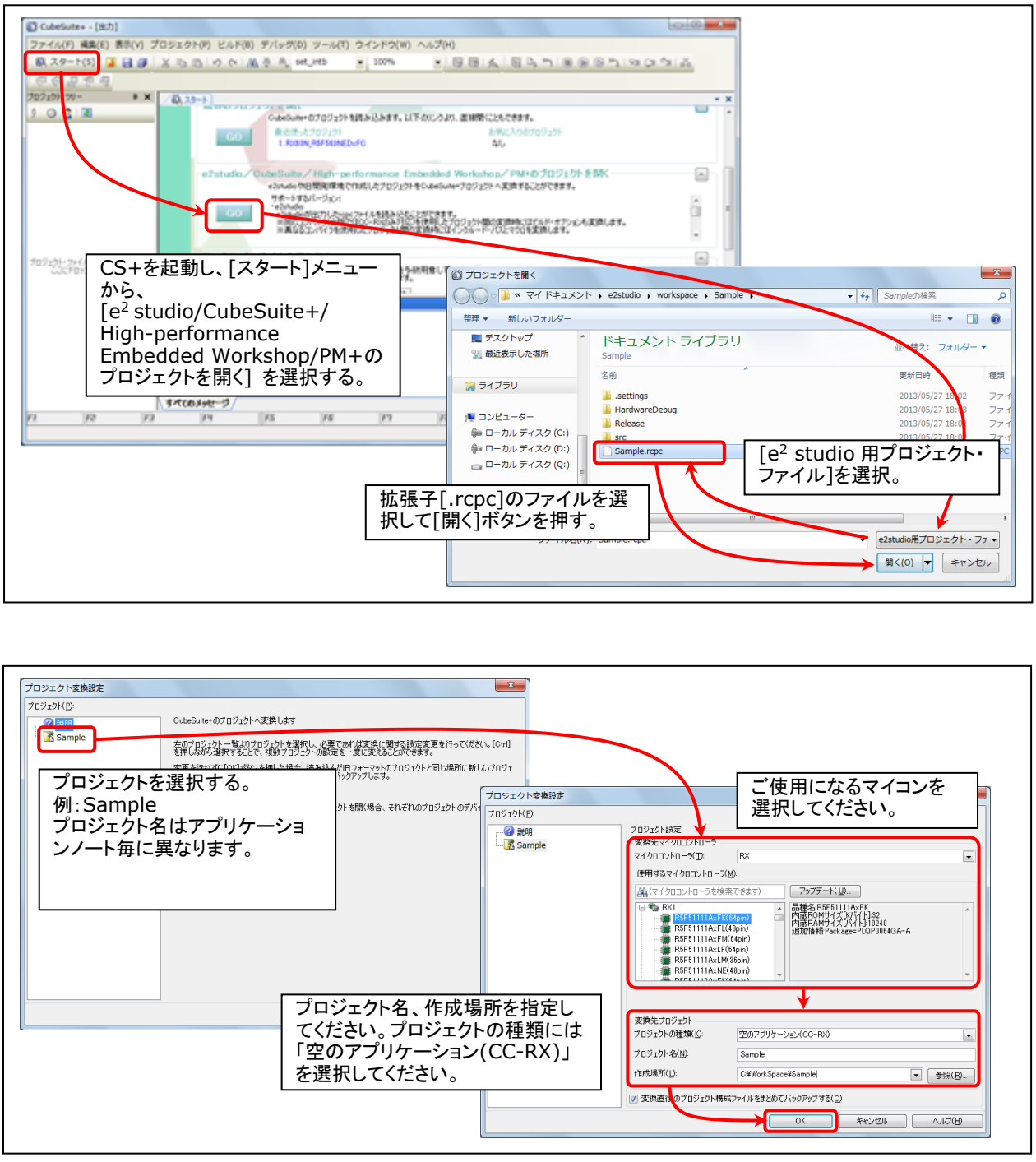


Figure 5-1 e² studio 用プロジェクトの CS+読み込み方法

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ
http://japan.renesas.com/

お問合せ先
http://japan.renesas.com/contact/

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Oct 16, 2014	—	初版発行
1.10	Dec 26, 2014	—	RX71M を対象デバイスに追加。
1.11	Sep 30, 2015	—	RX63N/RX631 を対象デバイスに追加。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は，製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、
   各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
   家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通用信号機器、
   防災・防犯装置、各種安全装置等
   当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。 たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。 なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
注2. 本資料において使用されている「当社製品」とは、注１において定義された当社の開発、製造製品をいいます。

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口 http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社　〒135-0061　東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：http://japan.renesas.com/contact/

© 2015 Renesas Electronics Corporation. All rights reserved.
Colophon 4.0